

Malik Tayeh

Nein oder nicht Nein - Die Mär von der bösen negativen Anforderung

Im Requirements Engineering Umfeld, gibt es seit längerem den Leitsatz alle seine Anforderungen positiv zu formulieren. Im Grunde naheliegend, denn schließlich beschreibt eine Spezifikation ja das Produkt das man erwartet und nicht das Produkt welches man nicht haben möchte. Dieser Leitsatz manifestiert sich z.B. im vielzitierten Standard IEEE 830-1998, der von der Verwendung von negativ formulierten Anforderungen abrät [1]. Meist wird dies mit der hohen Interpretierbarkeit von negativ formulierten Anforderungen und der schwierigen Testbarkeit dieser begründet, ohne allerdings klare Beispiele zu nennen, oder Fallstudien hierzu zu zitieren. Dieser Artikel beleuchtet die Frage inwiefern negativ formulierte Anforderungen tatsächlich problematisch sind.

Arten von negativ formulierten Anforderungen

Vorneweg gilt es zu klären was genau unter einer negativ formulierten Anforderung zu verstehen ist. Derer gibt es zwei Varianten die ich im Folgenden genauer erläutern möchte.

Die sogenannten inversen Anforderungen, sind negativ formulierte Anforderungen, die dem Zweck dienen, sowohl bestimmte Eigenschaften eines Produktes, Teile eines Prozesses, als auch bestimmte Funktionen in der fachlichen Verfeinerung oder in der technischen Realisierung explizit zu verbieten.

Ausgeschlossene Aspekte wiederum sind optionale, negativ formulierte Systemkontextinformationen. Sie dienen dem Zweck, mögliche Fehlinterpretationen bezüglich der Eigenschaften und Leistungen eines Produktes, eines Prozesses oder der am Prozess beteiligten Personen auszuschließen.

Eine inverse Anforderung ist demnach den klassischen, positiv formulierten Anforderungen gleichgestellt, wohingegen ein ausgeschlossener Aspekt keine Anforderung im eigentlichen Sinne ist, sondern meist eine Ergänzung von positiv formulierten Anforderungen darstellt, die dem Ziel dient, die Verständlichkeit dieser zu verstärken.

Vorteile inverser Anforderungen

Es stellt sich die Frage, warum man inverse Anforderungen überhaupt verwenden sollte. Schließlich lassen sich negativ formulierte Anforderungen meist in eine Positive Anforderung umformen. Aus einem „Das System darf nicht mehr als 5 Kg wiegen“, wird ein „Das System muss weniger als 5 Kg wiegen“. Durch eine simple Umformulierung haben wir eine verständliche und leicht testbare Anforderung gewonnen. Also eigentlich kein Bedarf für eine negative Formulierung.

Jeder der in seinem Berufsalltag regelmäßig Anforderungen schreibt, kennt aber auch die folgende Situation. Es liegt eine eigentlich gut verständliche, eindeutige Anforderung, wie z. B. die folgende vor. „Das Gehäuse des Systems darf keine allergischen Reaktionen auf Nickel auslösen“. Diese Anforderung lässt sich nicht positiv formulieren ohne den Sinn der ursprünglichen Anforderung zu verändern. Die Anforderung „Das Gehäuse des System muss nickelallergenfrei sein“, ist zwar positiv formuliert, verbietet gleichzeitig aber auch die Verwendung von Nickel im Gehäuse des Systems. Dieses Verbot war aber nie Teil der ursprünglichen Anforderung und schränkt nebenbei den Lösungsraum für den Entwickler unnötig ein. Denn die Verwendung von geringen Mengen Nickel im Gehäuse des zu entwickelnden Systems ist durchaus denkbar, vorausgesetzt der Anteil liegt unterhalb des Schwellenwertes der eine allergische Reaktion auslösen würde. Die Nennung dieses Schwellenwertes in unserer Anforderung, wie z.B. in „Das Gehäuse des System muss weniger als 0,05%¹ Nickel enthalten“, würde dieses Problem zwar lösen, bringt aber das Problem mit sich, dass zur Definition dieses Schwellenwertes sehr viel Know-How vorausgesetzt wird, welches wir eventuell nicht besitzen und ggf. erst langwierig recherchieren müssen.

Testbarkeit inverser Anforderungen

Da inverse Anforderungen – wie der Name schon sagt – ebenfalls Anforderungen sind, muss ihre Erfüllung natürlich getestet werden. Dies kann sich teilweise als sehr schwierig herausstellen und wird daher gerne als Argument für die Vermeidung von inversen Anforderungen herangezogen. Dies lässt sich aber über die Verfeinerung von inversen Anforderungen durch testbare, positiv formulierte Anforderungen lösen. „Verfeinerungen von Anforderungen geben eine Lösung für die abstrakteren Anforderungen vor.“ [2]. Wichtig ist, dass im Gegensatz zur Umformulierung von inversen Anforderungen in positiv formulierte Anforderungen,

die Verfeinerung die ursprüngliche inverse Anforderung nicht ersetzt, sondern nur ergänzt. Demnach bleibt der ursprüngliche Sinn der Anforderung unverändert und die Testbarkeit dieser wird dadurch gewährt, dass die neue Anforderung auf der tieferen (feineren) Spezifikationsebene positiv und testbar formuliert ist.

Beim Verfeinern von Anforderungen sollte man stets darauf achten, dass die Verfeinerung oftmals der erste Schritt in Richtung Lösungsdesign ist und je nach Spezifikationsebene eventuell gar nicht gewünscht ist. Sollte die Anforderung des Stakeholders tatsächlich nur sein, dass „das Gehäuse des Systems keine allergischen Reaktionen auf Nickel auslöst“, und es dem Stakeholder fachlich egal ist, ob dies durch ein zu 100% aus Aluminium bestehendes Gehäuse, ein Plastikgehäuse oder eine allergenfreie Legierung des Gehäuses umgesetzt wird, dann ist die ursprüngliche inverse Anforderung vollkommen ausreichend. Die Lösungsfindung und die Dokumentation der möglichen Lösungsvarianten obliegt dann dem Auftragnehmer und wird oftmals im vom Auftragnehmer erstellten Pflichtenheft dokumentiert und vom Auftraggeber abgenommen.

Ausgeschlossene Aspekte

Im Gegensatz zu inversen Anforderungen handelt es sich bei ausgeschlossenen Aspekten nicht um Anforderungen, sondern um ergänzende Informationen die einzig dem Ziel dienen die Verständlichkeit der Spezifikation zu erhöhen. Sie treten oftmals in Form eines Kommentars zu einer positiv formulierten Anforderung auf und sollen verhindern, dass die eigentliche Anforderung falsch interpretiert wird. Mögliche Anwendungsfälle für ausgeschlossene Aspekte wären z.B. die folgenden Situationen.

- Ein Altsystem wird abgelöst und eine existierende Funktionalität oder Eigenschaft soll in der neuen Implementierung nicht mehr vorhanden sein.
- Ein bisher manuell durchgeführter Prozess wird teilweise von einem neuen System abgelöst.
- Während vorgelagerten Diskussionen² waren Funktionalitäten oder Eigenschaften angedacht, die dann aber doch nicht umgesetzt werden sollen.
- In vorangegangenen Entwicklungszyklen ähnlicher Systeme war eine Funktionalität oder Eigenschaft vorhanden, die im aktuellen System nicht mehr vorhanden sein soll.
- Alle, oder fast alle Konkurrenzsysteme verfügen über eine Funktionalität oder Eigenschaft die im aktuellen System nicht vorhanden sein soll.

Formulierung

Um inverse Anforderungen innerhalb einer Spezifikation sauber von den ausgeschlossenen Aspekten zu trennen, sollten beide Typen bereits anhand ihrer Formulierung zu unterscheiden sein. Hierzu empfiehlt es sich – genauso wie bei positiv formulierten Anforderungen – das Modalverb der rechtlichen Verbindlichkeit als Indikator zu nutzen. Die gängigen Modalverben „muss“, „sollte“ und „wird“ [1], [2], [3] erweitern wir daher um ein viertes Modalverb, nämlich „darf nicht“ bzw. „darf kein“. Hiermit wird bereits anhand der Formulierung klar, dass es sich um eine Anforderung handelt, und dass diese ein klares Verbot einer Eigenschaft oder einer Funktionalität beschreibt. Bei ausgeschlossenen Aspekten ist die Verwendung von Modalverben der rechtlichen Verbindlichkeit – ähnlich wie bei allen anderen Spezifikationsinhalten die keine Anforderung sind – zu vermeiden.

Fazit

Die Verwendung von negativen Formulierungen in Spezifikationen kann durchaus Vorteile bieten. Man vermeidet die unbeabsichtigte Einschränkung von Lösungsmöglichkeiten mithilfe von inversen Anforderungen und kann die Verständlichkeit von positiv formulierten Anforderungen durch die Dokumentation von ausgeschlossenen Aspekten verbessern.

Textanmerkungen

¹ Dieser Schwellenwert ist nur ein Beispiel.

² z.B. während der Analysephase, in Workshops oder während der Erstellung eines Angebotes

Referenzen

[1] IEEE Standards Board: IEEE Std 830-1998. IEEE Recommended Practice for Software Requirements Specification. IEEE Press, 1998

[2] Rupp, Chris; SOPHISTen: Requirements- Engineering und Management – Professionelle iterative

Anforderungsanalyse für die Praxis, Hanser, Nürnberg 2009

[3] Pohl, Klaus; Requirements Engineering – Grundlagen, Prinzipien, Techniken; dpunkt; Heidelberg, 2009

Copyright © 2014 by SOPHIST GmbH

Publikation urheberrechtlich geschützt. Alle Rechte, auch die der Übersetzung, des Nachdruckens und der Vervielfältigung oder Teilen daraus, vorbehalten. Kein Teil der Publikation darf in irgendeiner Form, egal welches Verfahren, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet werden, vervielfältigt oder verbreitet werden.

Dies gilt auch für Zwecke der Unterrichtsgestaltung. Eine schriftliche Genehmigung ist einzuholen. Die Rechte Dritter bleiben unberührt.